WOA7015 Advanced Machine Learning
Universiti Malaya
2024/25 Sem1

Group Assginment 3 (20%)

1 Instruction

In today's lecture, you have learned about multilayer perceptron (MLP), backpropagation, and techniques on how to overcome overfitting. You are given a fetus dataset where the first column is the outcome and the second to last column are the features. You have a total of 16 features.

2 Your Tasks

Find a group of 2-3 students. Build a 2-layer MLP for fetus classification. Feel free to play around with the model architecture and see how the training time/performance changes, but to begin, try the following:

Input (16 dimensions) -> fully connected layer (10 hidden units) -> nonlinearity (ReLU) -> fully connected (5 hidden units) -> softmax

Some hints:

Even as we add additional layers, we still only require a single optimizer to learn the parameters. Just make sure to pass all parameters to it!

This MLP model has many more parameters than the logisitic regression example, which makes it more challenging to learn. To get the best performance, you may want to scale your datasets, play with the network architecture, learning rate and increase the number of training epochs. If you are scaling the features, make sure the scaling is according to the training data distribution to avoid data leakage.

Be careful using torch.nn.CrossEntropyLoss(). If you look at the PyTorch documentation: you'll see that torch.nn.CrossEntropyLoss() combines the softmax operation with the cross-entropy. This means you need to pass in the logits (predictions pre-softmax) to this loss. Computing the softmax separately and feeding the result into torch.nn.CrossEntropyLoss() will significantly degrade your model's performance!

3 Evaluation

1. Code (10%)

   (a) 2% - Instantiate two Datasets – one for training and one for testing. Create two instances of the DataLoader class by passing in the train and testing datasets. Specify the batch size using the batch size argument. Make sure to use shuffle=True for the train loader!

   (b) 2% - Instantiate your MLP class with init and forward methods.

   (c) 2% - Evaluate your model performance with testing data. Save the training loss, testing loss, training accuracy, and testing accuracy at every epoch.

   (d) A working model should have a testing accuracy of up to 65%. If your model doesn't work quite yet, it may be because of the hyperparameters. Play around with the batch size, MLP hidden size, MLP hidden layer, and learning rate to maximize your testing accuracy.

   (e) 4% Hyperparameter - The performance of neural networks is highly dependent on hyperparameters: these are values that are manually set by the ML engineer rather than learned as parameters that are part of the model. For our model, the hyperparameters include: learning rate, activation function, number of layers, loss function, and regularization. It is often stated in deep learning that choosing good hyperparameters is an art more than a science. ML engineers typically use their prior experience to narrow down the hyperparameters to a reasonable range before conducting some sort of search. This assignment is the beginning of that experience for you!

   Two commonly used search policies are:

   i. Random: we simply choose a random value wthin some range [a,b] for continuous hyperparameters or from some set {a,b,...} for categorical hyperparameters. We repeat until the model achieves satisfactory performance.

   ii. Gridsearch: for continuous hyperparameters, we start at the lower bound a and increase the value of the hyperparameter by some increment i until upper bound b. Finally we take the value that gave us the best performance. For categorical hyperparameters, we try all possible values and take the best one. This may not be feasible, given the number of possibilities and the time available.

2. Report (10%)

   (a) 6% - Result tables tabulating the testing accuracy using different hyperparameters. Comment on what you observe and what the results mean. Which model has the best results - state the hyperparameter used.

   (b) 3% - Using the best model configuration, plots the following plots

      i. 1 plot with training loss, testing loss across epochs

      ii. 1 plot with training accuracy, testing accuracy across epochs

   Comment on what you observe and what the results mean? Does your model have an overfitting issue?

(c) 1% - Effectiveness of your figure and table. Every figure and table should have a caption. The caption for a figure appears below the graphic; for a table, above. Captions should be brief, yet provide a comprehensive explanation of the data as it appears within the text. Refer to Useful Links No.5 to learn how to write a good caption.

(d) Member contributions

4 Submission

One group submits only one set of the following file.

1. Source code (.ipynb) (10%)
2. Report (in pdf) (10%)
   (a) First Page - Title, Group Members' Name and Matrix No.
   (b) Font - 12pt, Times New Roman
   (c) Double Spacing, single column
   (d) Max pages - 5 pages excluding the first page
   (e) If you are referring to any source, please cite and compile them in Reference. You can use Endnote to help with your reference formatting. You can install Endnote from UM software website. Refer to Useful Link No.6-7 on how to insert reference in Words and download citation from Google Scholar.

5 Useful Links

1. https://machinelearningmastery.com/training-a-pytorch-model-with-dataloader-and-dataset/
2. https://averdones.github.io/reading-tabular-data-with-pytorch-and-training-a-multilayer-perceptron/
3. https://medium.com/@shashankshankar10/introduction-to-neural-networks-build-a-single-layer-perceptron-in-pytorch-c22d9b412ccf
4. https://medium.com/@reddyyashu20/deep-learning-with-pytorch-780be96b2819
5. https://www.enago.com/academy/practical-tips-figure-tables-legends-for-manuscripts/
6. https://www.youtube.com/watch?v=-EeyfQ79Rlw&ab_channel=CurtinLibrary
7. https://uri.libguides.com/google/gscholexport